6

```
# Import required libraries
from sklearn import datasets
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.svm import SVC
from sklearn.metrics import classification_report, confusion_matrix
import matplotlib.pyplot as plt

# Load MNIST digits dataset
digits = datasets.load_digits()

# Features and labels
X = digits.data
y = digits.target

# split into training & testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size
=0.2, random_state=42, stratify=y)

# Feature scaling
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

# Train the SVM model
svm_model = SVC(kernel='rbf', gamma=0.001, C=10)
svm_model.fit(X_train_scaled, y_train)
```

```
#Make predictions
y_pred = svm_model.predict(X_test_scaled)

#Evaluation
print("Classification Report:\n", classification_report(y_test,
  y_pred))
print("Confusion Matrix:\n", confusion_matrix(y_test, y_pred))
#Display some predicted digits
plt.figure(figsize=(10,4))
for index, (image, prediction) in enumerate(zip(X_test[:8],
  y_pred[:8])):
    plt.subplot(2,4, index+1)
    plt.imshow(image.reshape(8,8), cmap=plt.cm.gray_r)
    plt.title(f"Pred: {prediction}")
    plt.axis('off')
plt.tight_layout()
plt.show()
```

6th prg output

Classification Report:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	36
1	0.95	0.97	0.96	36
2	1.00	1.00	1.00	35
3	1.00	1.00	1.00	37
4	0.97	1.00	0.99	36
5	1.00	1.00	1.00	37
6	1.00	1.00	1.00	36
7	0.97	1.00	0.99	36
8	1.00	0.94	0.97	35
9	1.00	0.97	0.99	36
accuracy			0.99	360
macro avg	0.99	0.99	0.99	360
weighted avg	0.99	0.99	0.99	360

Confusion Matrix:

```
[[36  0  0  0  0  0  0  0  0  0]
 [ 0 35  0  0  1  0  0  0  0  0]
 [ 0  0 35  0  0  0  0  0  0  0]
 [ 0  0  0 37  0  0  0  0  0  0]
 [ 0  0  0  0 36  0  0  0  0  0]
 [ 0  0  0  0  0 37  0  0  0  0]
 [ 0  0  0  0  0  0 36  0  0  0]
 [ 0  0  0  0  0  0  0 36  0  0]
 [ 0  2  0  0  0  0  0  0 33  0]
 [ 0  0  0  0  0  0  0  1  0 35]]
```